

Biotechvana Queue Manager 1.0

Futami, R.¹ Lloréns, C.^{1,2} and Moya, A.^{2,3}

1 - Biotechvana, Valencia, Spain

2 - Instituto Cavanilles de Biodiversitat i Biología Evolutiva Universitat de València, Spain

3 - CIBER de Epidemiología y Salud Pública (CIBERESP), Spain

Corresponding author: ricardo.futami@biotechvana.com

Availability: Available online February 1, 2008. Biotechvana Queue Manager is distributed under the terms of the Biotechvana Open Source License (URL 1).

Biotechvana Queue Manager is a server overload preventing script for PHP-based Web sites. This script manages all requests received in a web server to a target application. The tool registers them in a queue controlled by a database engine and processes each request sequentially, one at a time. It operates transparently, telling the user how much time remains for a request to be performed.

Keywords: PHP | queues | server load balancing | server overload prevention

OVERVIEW

Offering online web services for public access in our server can lead to a degradation of the service provided when too many users access them simultaneously, especially when these services are high resource consumers. Thus, a queue management utility becomes essential as a safeguard against overloading the server (for a more information in this topic see 1). Overloading can lead to low performance, or even block the server to client requests. Biotechvana Queue Manager (BQM) functions as an intermediate layer between the application's web interface, usually a form, and the target script that processes the request and executes the target script. BQM captures user's parameters defined in the form, and assigns the user request an identification code. This code is entered in a queue, which is managed using the FIFO (First In First Out) queuing strategy, or 'what comes in first is handled first'. Once the identification code reaches the first position in the queue, the user's request is sent to the processing script that runs the target application. While the request is in the queue, the user is redirected to a wait page that displays the estimated time remaining to process the request. As shown in Figure 1 BQM workflow can be represented as four separated layers:

1) Web form layer allows the user to input data in the target application. The form action field must be set pointing to BQM

2) BQM layer receives all the input from the first layer and stores it until sending it to the execution layer. The user's request is registered with a unique identifier, which is entered in a queue of pending requests in our database. If there is any preceding request in the queue, the user is redirected to a wait page that displays the time remaining for its task to be executed. This time is estimated from the request's placement in the queue and the complexity of the task to be executed. Once the waiting time has elapsed, the user is redirected to BQM to

check if it can be executed. When a process reaches the top of the queue it is removed, and the user's full input is sent to the execution layer.

3) Execution layer: The processing script receives all data from the initial form and executes the target application.

4) Results layer: Results are presented to the user.

INSTALLATION

First, confirm that a Web server engine, a PHP application server, and a DBMS are properly installed on your system. You can download and install a Web server like Apache at URL 2 for Windows and Linux platforms; or an IIS (Internet Information Services) Web server for Windows platforms, which comes included in Windows server versions. Next, install the PHP application server, which can be downloaded at URL 3; Installation instructions are provided on its corresponding web sites. Finally, install the MySQL DBMS. Many DBMS are available, such as MySQL at URL 4, Firebird at URL 5, or PostgreSQL at URL 6. A list of databases supported by this application is available at the AdoDB Web site URL 7. This script has been successfully tested in a MySQL DBMS; it is thus the choice we recommend. Once a Web server engine, the PHP application server, and a DBMS are properly installed and working, proceed with the installation of BQM, as described below.

a) Unpack the zip file and copy the following files to the folder that contains your web application:

- Queue_manager.php
- Database.php
- wait.php
- create_database.bat
- create_database.sql

b) Open the file 'Queue_manager.php' and search for the following line to define the location of the target-processing script of your service:

```
define('ACTION_PAGE', 'test/testparse.php');
```

c) Open the file 'Database.php', and edit these lines with your own database configuration:

```
var $user = 'queue-manager'; → DBMS user name
var $pass = 'queue'; → DBMS user password
var $host = 'localhost'; → DBMS host
var $engine = 'mysql'; → DBMS engine name
var $database = 'queue-manager_queue'; → database name
```

d) Edit your Web form and set the action of the form tag pointing to 'Queue_manager.php'.

```
<form name='myform' action='Queue_manager.php'
method='post'>
.....
</form>
```

All the data contained in your form will be received by Queue_manager.php, registered in the queue and ultimately sent to the target service.

REQUIREMENTS

To install BQM, it is required a PHP application server, a web server, and a database management system (DBMS). A web server is a computer program responsible for accepting HTTP requests from a user's web browsers and delivering contents such as HTML web pages, images, and other files. An application server is software that helps a web server to process Web pages containing server-side scripts that cannot

be processed by a regular web server engine. When a dynamic page is requested by a visitor's browser, the web server calls the application server for processing the scripts before sending the page to the browser. A DBMS is computer software designed for the purpose of managing databases. It controls the organization, storage, management, and retrieval of data in a database. Some knowledge of HTML language is also required, principally in Web forms management, the POST and GET methods, and URL parameter codification.

ACKNOWLEDGMENTS

We thank Rachel Epstein for language revision and the Servei Central de Suport a la Investigació Experimental (SC-SIE) (URL 8) at UVEG for technical support. Biotechvana Bioinformatics has been awarded the NOVA 2006 by IMPIVA and Conselleria d'Empresa, Universitat I Ciència of Valencia. The research has been partly supported by grants IMCB-TA/2005/45, IMIDTD/2006/158 and IMIDTD/2007/33 from IMPIVA, and by grant BFU2005-00503 from MEC to AM.

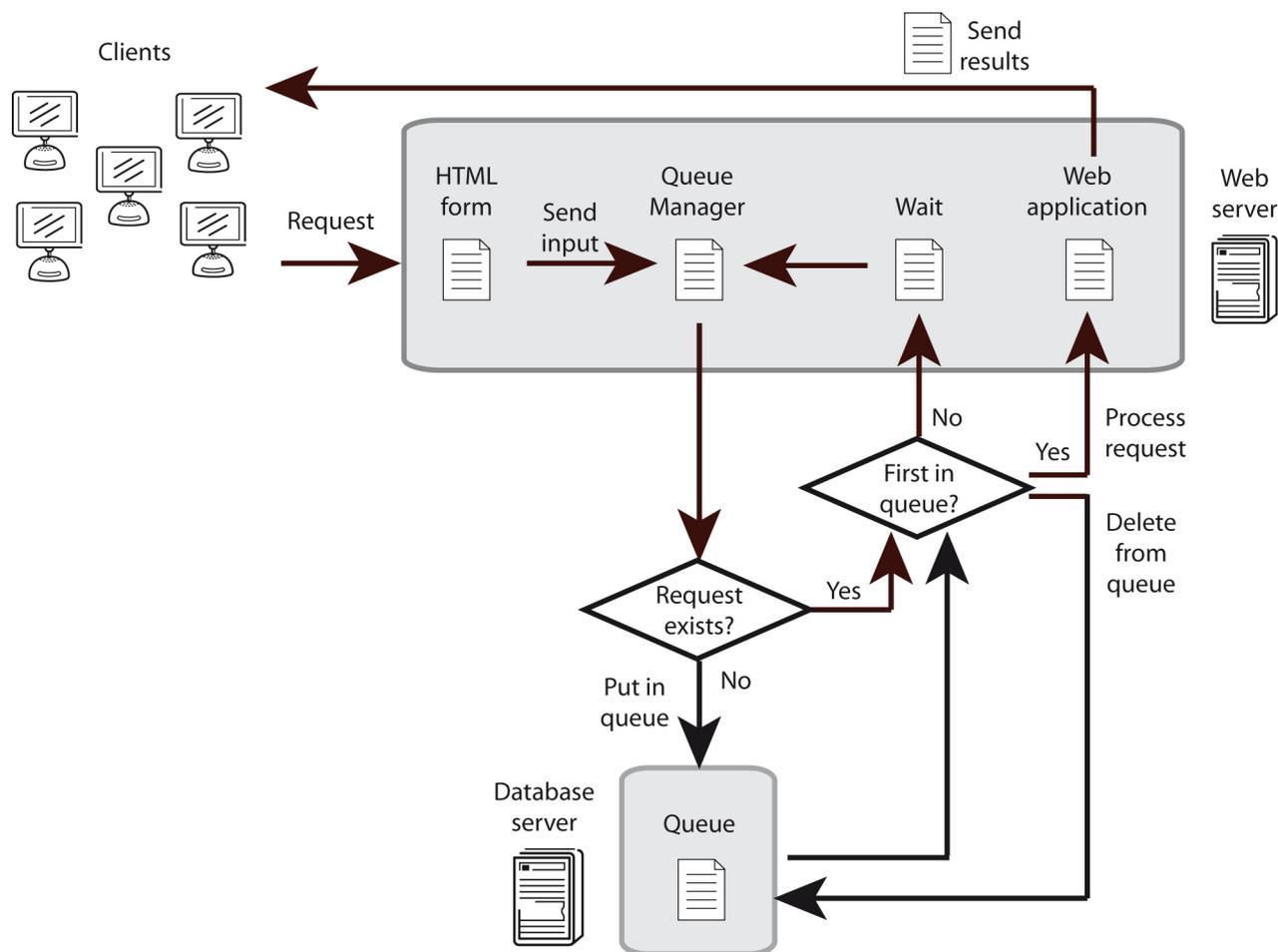


Figure 1. Queue Manager flow diagram.

LITERATURE

1. Gross, D., and C. M. Harris (1998). Wiley Series in Probability and Statistics, Wiley-Interscience; 3 Subedition.

URLS

1. **Open Source License:** http://biotechvana.com/loader.php?section=contents&page=terms_ocl
2. **PHP programming language:** <http://php.net>
3. **Apache HTTP server:** <http://www.apache.org>
4. **MySQL Database Management System:** <http://www.mysql.com>
5. **Firebird Database Management System:** <http://www.firebirdsql.org>
6. **Postgres Database Management System:** <http://www.postgresql.com>
7. **AdoDB database abstraction library:** <http://adodb.sourceforge.net>
8. **SCSIE, Universitat de València:** <http://scsie.uv.es>

SPONSORS



VNIVERSITAT DE VALÈNCIA



UNIÓN EUROPEA
Fondo Europeo de Desarrollo Regional
Una manera de hacer Europa